NASA/GSFC's Flight Software Core Flight System

David McComas

Flight Software Systems Branch

NASA/Goddard Space Flight Center

Flight Software Workshop

November 7-9, 2012

Southwest Research Institute

San Antonio, Texas

Agenda

1. CFS Background
2. Customers
3. Architecture & Component Summary
4. Example Usage
5. Next Steps
6. Conclusion

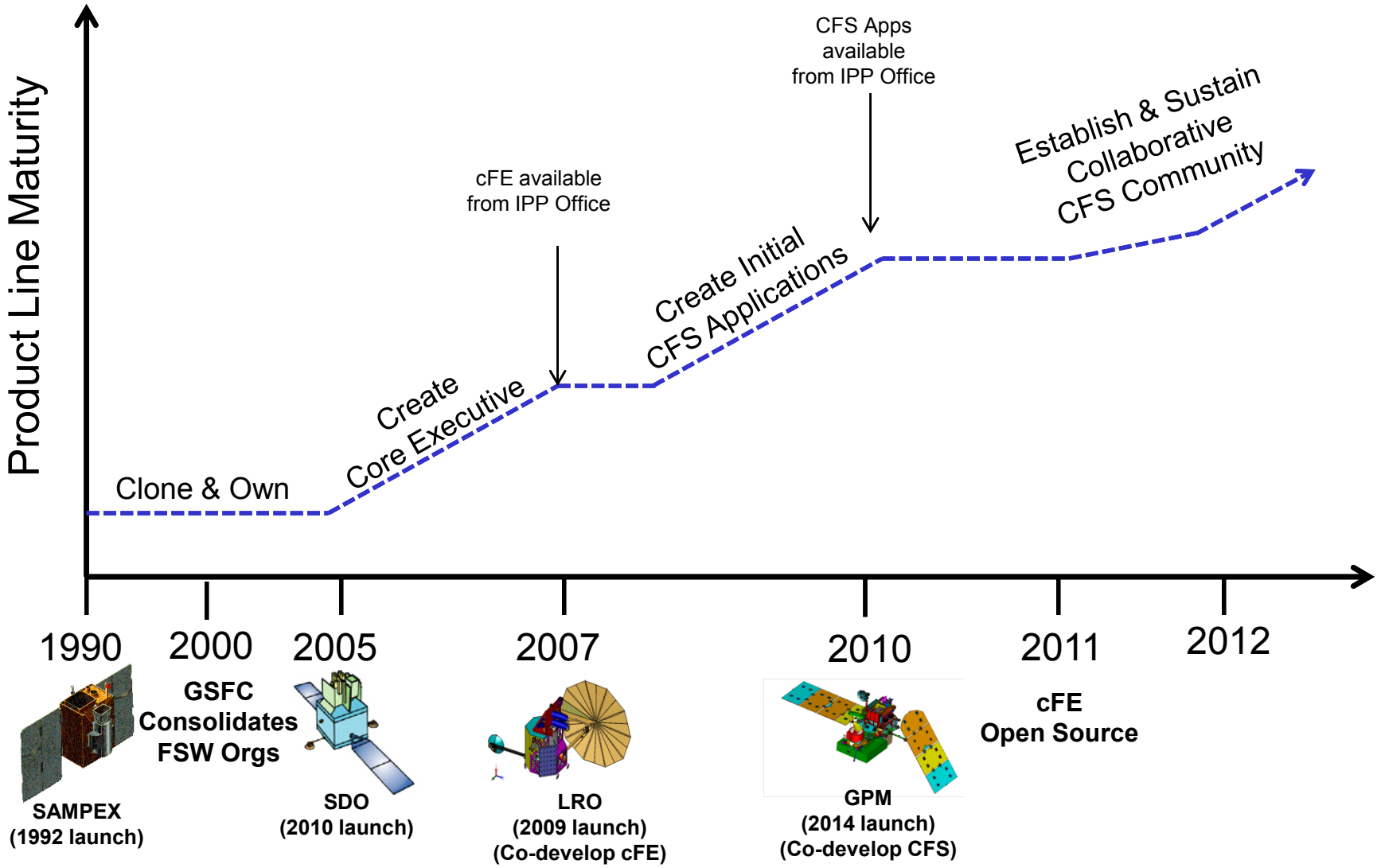
CFS Background

- **After 15 years and over 10 missions of "clone and own" Goddard's FSW Systems Branch recognized the need to develop a product line approach towards FSW**
  - Smaller budgets and shorter schedules demanded a change

- **CFS has been a grass roots effort with limited funding**
  - Architecture and components have evolved in parallel with projects in an opportunistic manner
  - It was not a top-down approach in the sense that the product line was required to cover a specific suite of processors, operating systems, meet performance requirements, etc.
    - The CFS lab is very limited and the CFS scope is expanded through projects
  - It was a methodical engineering approach and a Heritage Analysis was performed on each component

- **We are at a critical strategic juncture for the CFS**
  - The cFE went open source in the fall of 2011 and has started gaining momentum
  - Most NASA centers have requested the cFE
  - Within the past two months Code 582 has been contacted by 2 commercial ventures and the Korean Aerospace Research Institute

CFS Timeline



cFE/CFS Customers

Organization	Purpose	Products	Notes
NASA / GSFC	LRO Spacecraft	*cFE*	Co-developed the core Flight Executive
NASA / GSFC	GPM Spacecraft	cFE, *CFS*	Co-developed the Core Flight System applications
NASA / GSFC	MMS Spacecraft	cFE, CFS	Ported cFE to a Coldfire RTEMs platform
NASA / GSFC	ATLAS Instrument	OSAL, cFE	Instrument using OSAL on a RAD750, RTEMs platform Simulator using cFE
JHU / APL	RBSP Spacecraft	cFE	Using the cFE without table services
JHU / APL	Solar Probe Plus Spacecraft	cFE, CFS?	Ported cFE to SPARC/LEON3 RTEMS platform
NASA / ARC	LADEE Spacecraft	cFE, CFS	Created Simulink Interface Layer (SIL)
NASA / JPL	ESTO	cFE, CFS	
NASA / JSC	Project Morpheus	cFE, CFS	Closed-loop desktop simulation system
NASA / KSC, MSFC, GRC	Evaluation	cFE, CFS	
Fraunhofer CESE	Research	cFE	Architecture evaluation and testing techniques
NASA/GSFC, JHU/APL	Multiple IRADs	cFE	Memory Protection, Multi-Core, Virtualization

cFE Open Source Downloads

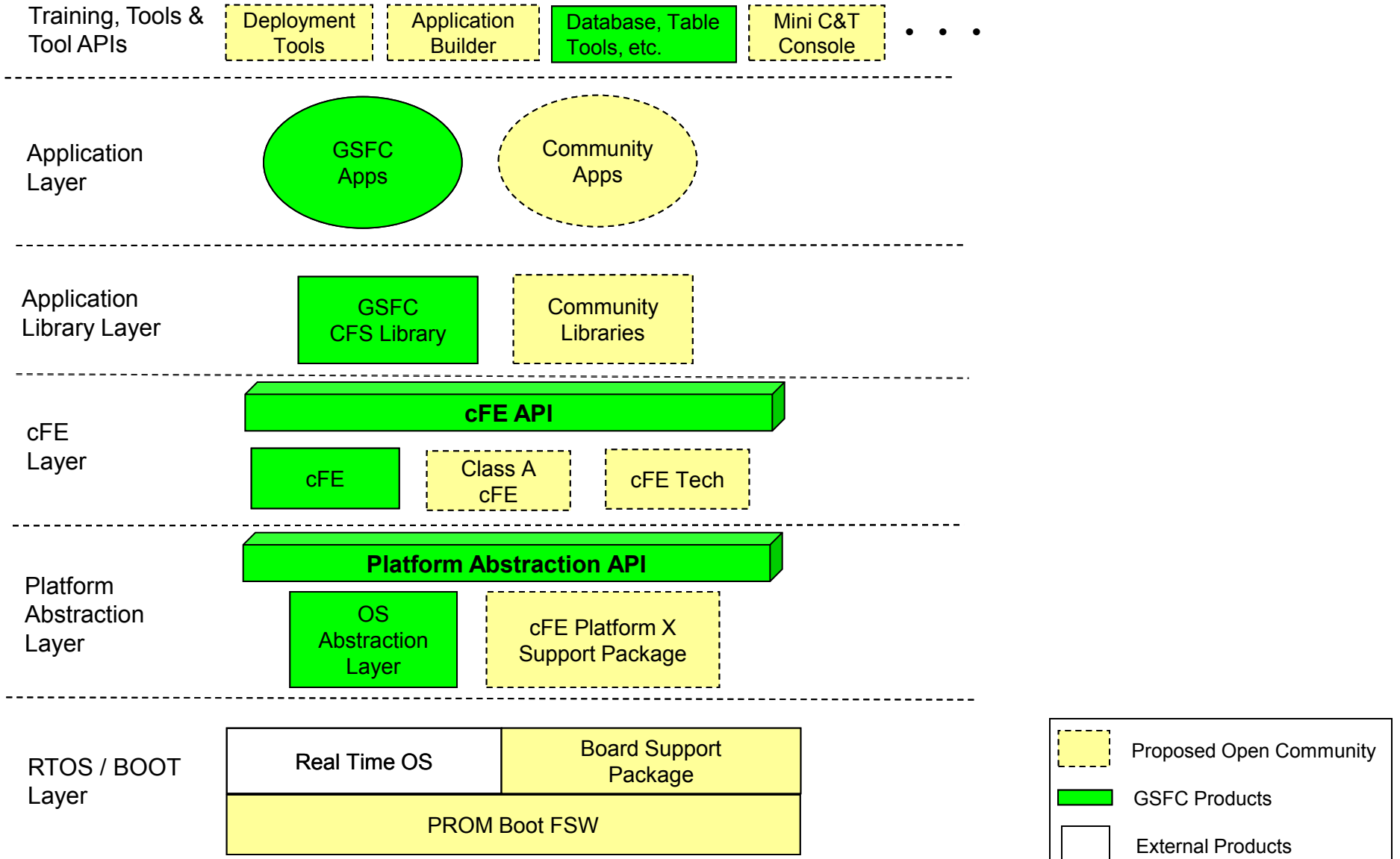
	Country	BSD	Linux	Macintosh	Unknown	Windows	Total
1.	United States	0%	13%	7%	1%	29%	578
2.	India	0%	10%	0%	0%	48%	29
3.	Germany	4%	12%	4%	0%	12%	26
4.	United Kingdom	0%	5%	27%	0%	14%	22
5.	France	0%	14%	0%	0%	0%	21
6.	Canada	10%	0%	10%	0%	20%	20
7.	China	0%	5%	10%	5%	60%	20
8.	Brazil	0%	6%	0%	0%	50%	18
9.	Korea	0%	29%	0%	0%	12%	17
10.	Russia	0%	0%	0%	0%	6%	16
11.	Netherlands	0%	0%	7%	0%	21%	14
12.	Argentina	0%	0%	0%	0%	54%	13
13.	Australia	0%	23%	23%	0%	0%	13
14.	Spain	0%	8%	8%	0%	42%	12
15.	Finland	0%	50%	10%	0%	30%	10
16.	Austria	0%	0%	0%	0%	0%	9
17.	Lithuania	0%	0%	0%	0%	0%	8
18.	Turkey	0%	0%	29%	0%	57%	7
19.	Italy	0%	0%	0%	0%	57%	7
20.	Poland	0%	0%	0%	17%	33%	6
21.	Sweden	0%	33%	17%	0%	0%	6
22.	Bulgaria	0%	33%	0%	0%	17%	6
23.	Portugal	0%	0%	0%	0%	50%	6
24.	Ireland	0%	20%	0%	0%	0%	5
25.	Norway	0%	0%	0%	0%	0%	4
26.	Belgium	0%	25%	0%	0%	25%	4
27.	South Africa	0%	0%	0%	0%	50%	4
28.	Taiwan	0%	0%	0%	0%	25%	4

CFS Architecture

- **CFS uses a layered architecture**
  - Each layer “hides” its implementation and technology details.
  - Internals of a layer can be changed -- without affecting other layers’ internals and components.
  - Enables technology infusion and evolution.
  - Doesn’t dictate a product or vendor.
  - Provides Middleware, OS and HW platform-independence.

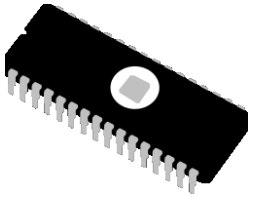
- **Positive feedback from two architecture reviews**
  - In February 2012 the NASA Software Architecture Review Board (SARB) published their assessment of the CFS
    - *“…Given these qualities, cFE/CFS has the potential to become one of the dominant architecture frameworks for NASA flight software (and simulation and test software).”*
  - Fraunhofer Center for Experimental Software Engineering (CESE) analyzed the consistency between the documented CFS architectural rules and the CFS implementation
    - “This paper analyzed the CFS product line architecture by verifying that architectural rules related to the module architecture and the code architecture were indeed met in the implementation..”

CFS Architecture



RTOS / Boot Layer

- **PROM Boot Software**
  - PROM resident software that does early initialization and bootstraps the Operating System
  - Provides ground based EEPROM/Flash loader
  - Keep it as simple as possible to minimize PROM changes
  - Commonly used Boot Software
    - RAD750 – BAE SUROM
    - Coldfire – Custom GSFC developed
    - LEON3 – uBoot – or Gaisler MKPROM



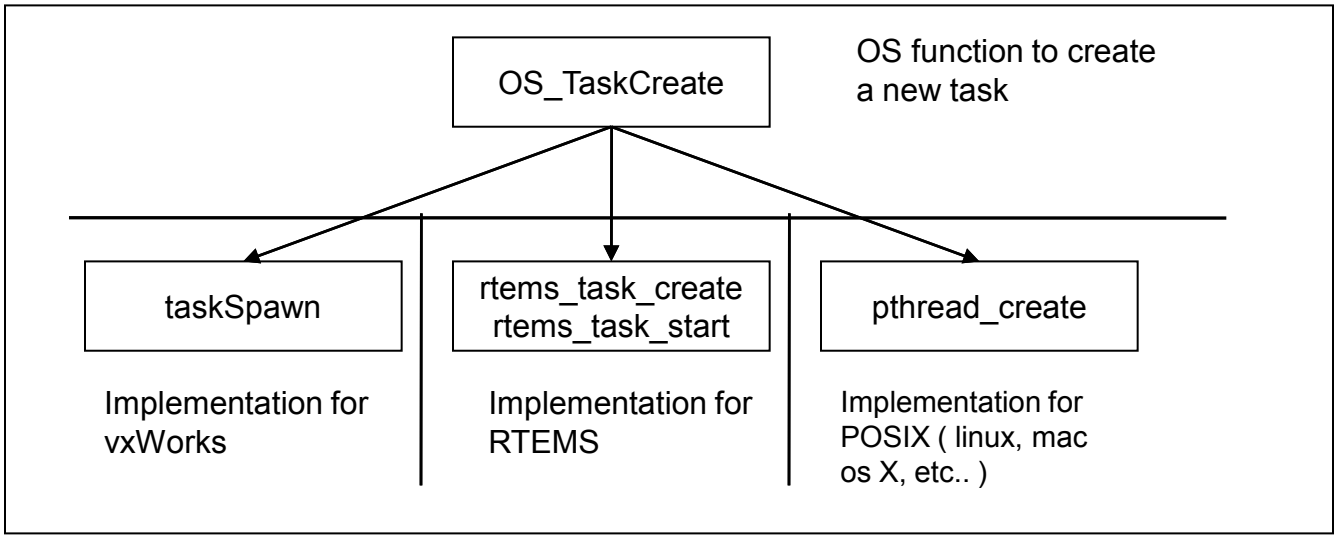
- **Real Time Operating System**
  - Pre-emptive priority based multi-tasking
  - Message Queues, Semaphores
  - Interrupt handling, Exception Handling
  - File systems, and shell
  - Supported Real Time Operating Systems
    - VxWorks
    - RTEMS
    - Linux (Not real time, but used for desktop development)

Embedded With RTEMS www.rtems.com

VxWorks

Platform Abstraction Layer - OSAL

- With the Platform Abstraction Layer, flight software such as the Core Flight Executive can run on multiple operating systems without modification
- The Operating System Abstraction layer (OSAL) is a small software library that isolates our Flight Software from the Real Time Operating System
- Current Implementations of the OSAL include:
  - RTEMS - Used on the RadHard Coldfire 5208, RAD750, LEON3
  - vxWorks - Used on RAD750
  - Linux / x86 - Used to run software on Desktop PC with Linux



Platform Abstraction Layer - PSP

- **Platform Support Package (PSP)**
  - A Platform Support Package is all of the software that is needed to adapt the cFE Core to a particular OS and Processor Card.
  - A Platform Support Package also includes all of the tool chain's specific make rules and options
  - Each mission is expected to customize a Platform Support Package
- **Functions included**
  - Startup code
  - EEPROM rand Memory read, write, copy, and protection functions
  - Processor card reset functions
  - Exception handler functions
  - Timer functions
- **Available PSPs**
  - Desktop Linux for prototyping
  - Power PC MCP750 / RAD750 – VxWorks 6.x
  - Coldfire - RTEMS

core Flight Executive (cFE)

- **A set of *mission independent, re-usable, core* flight software services and operating environment**
  - Provides standardized Application Programmer Interfaces (API)
  - Supports and hosts flight software applications
  - Applications can be added and removed at run-time (eases system integration and FSW maintenance)
  - Supports software development for on-board FSW, desktop FSW development and simulators
  - Supports a variety of hardware platforms
  - Contains platform and mission configuration parameters that are used to tailor the cFE for a specific platform and mission.
  - Provides 5 core services

1. Executive Services

- Manages startup: Power-On, Processor, Application resets, Child task spawning
  - Application and child task record keeping
  - Applications can be added and removed at run-time (eases system integration and FSW maintenance)
- Manages a system log for capturing reset and exception information
- Provides Critical Data Store (CDS)
  - Restores critical data after processor resets
- Provides ability to load shared libraries
- Provides support for device drivers
- Provides Performance Analysis support

core Flight Executive (cont)

2. **Event Services**
   - Provides application interface for sending asynchronous time-stamped debug/informational/error messages
   - Supports type-based and frequency filtering
   - Provides optional local (to processor) event log
3. **Software Bus Services**
   - Provides a portable inter-application message service
   - Provides runtime publish/subscribe software bus messaging interface
   - Reports errors detected during message transfers
   - Outputs statistics packets and routing information when commanded
4. **Table Services**
   - Tables are named groups of parameters and cFE provides interface to manage table images
   - Table Registry is populated at run-time eliminating cross-coupling of applications with flight executive at compile time
   - Performs table updates synchronously with application to ensure table data integrity
   - Tables can be shared between applications
   - Allows non-blocking table updates in Interrupt Service Routines (ISRs)
5. **Time Services**
   - Provides spacecraft time (derived from mission elapsed time MET), a spacecraft time correlation factor (STCF), and optionally, leap seconds
   - Provides interface to query time
   - Distributes “1Hz wakeup” and “time at the tone” messages

Application Library Layer

- **Useful for Mission Specific shared libraries**
  - MMS : Interfaces to the FPGA, CFDP buffer system
- **CFS Library provides common utility functions:**
  - CRC computations
  - String manipulation
  - File path and name validation
- **Missions can submit useful generic functions back to the CFS library**

Application Layer

- **Currently 11 FSW applications in GSFC library (fully qualified)**
  - Additional applications have been used in lab settings

1. CFDP (CF)

- Implements the CCSDS File Delivery Protocol
- Designed with a CFDP “engine” and an application layer
- Complicated application that requires more work. GPM and MMS have diverged

2. Checksum (CS)

- Performs checksum across static code/data regions specified by the users and reports errors
- Typically scheduled to run on a 1Hz schedule
- Byte-limited per execution cycle to prevent CPU hogging

3. Data Storage (DS)

- Stores Software Bus messages (packets) to data storage files according to destination table definition
- Filters packets according to packet filter table definition

Application Layer (cont)

4. File Manager (FM)

- Manages onboard files
  - Copy, Move, Rename, Delete, Close, Decompress, and Concatenate files providing file information and open file listings
- Manages onboard directories
  - Create, delete, and providing directory listings
- Reports free space on a per device basis

5. Health and Safety (HS)

- Monitor application execution and take table-defined action if hung
- Monitor event messages and take table-defined fro specific events
- Manage processor watchdog

6. Housekeeping (HK)

- Build combined telemetry messages containing data from multiple messages
- Notify ground when expected data is not received

Application Layer (cont)

7. Limit Checker (LC)

- Monitors table-defined Watch Points (WP) – Compare SB message data against constant threshold value
- Evaluate table-defined Action Points (AP) – Logical combinations of WPs using Reverse Polish Notation
  - Execute stored commands in response to tripped APs

8. Memory Dwell (MD)

- Augment telemetry stream by creating new packets with table-defined data points
- Data points can be any accessible memory locations

9. Memory Manager (MM)

- Perform memory read and writes (peek and poke) operations
- Perform memory loads and dumps from/to files

Application Layer (cont)

10. Scheduler (SCH)

- Operates a Time Division Multiplex (TDM) schedule of applications vie SB messages
  - Synchronized to external Major Frame (typically 1Hz signal)
  - Major Frame split into platform configured slots with a configured number of messages that can be sent for the slot

11. Stored Command (SC)

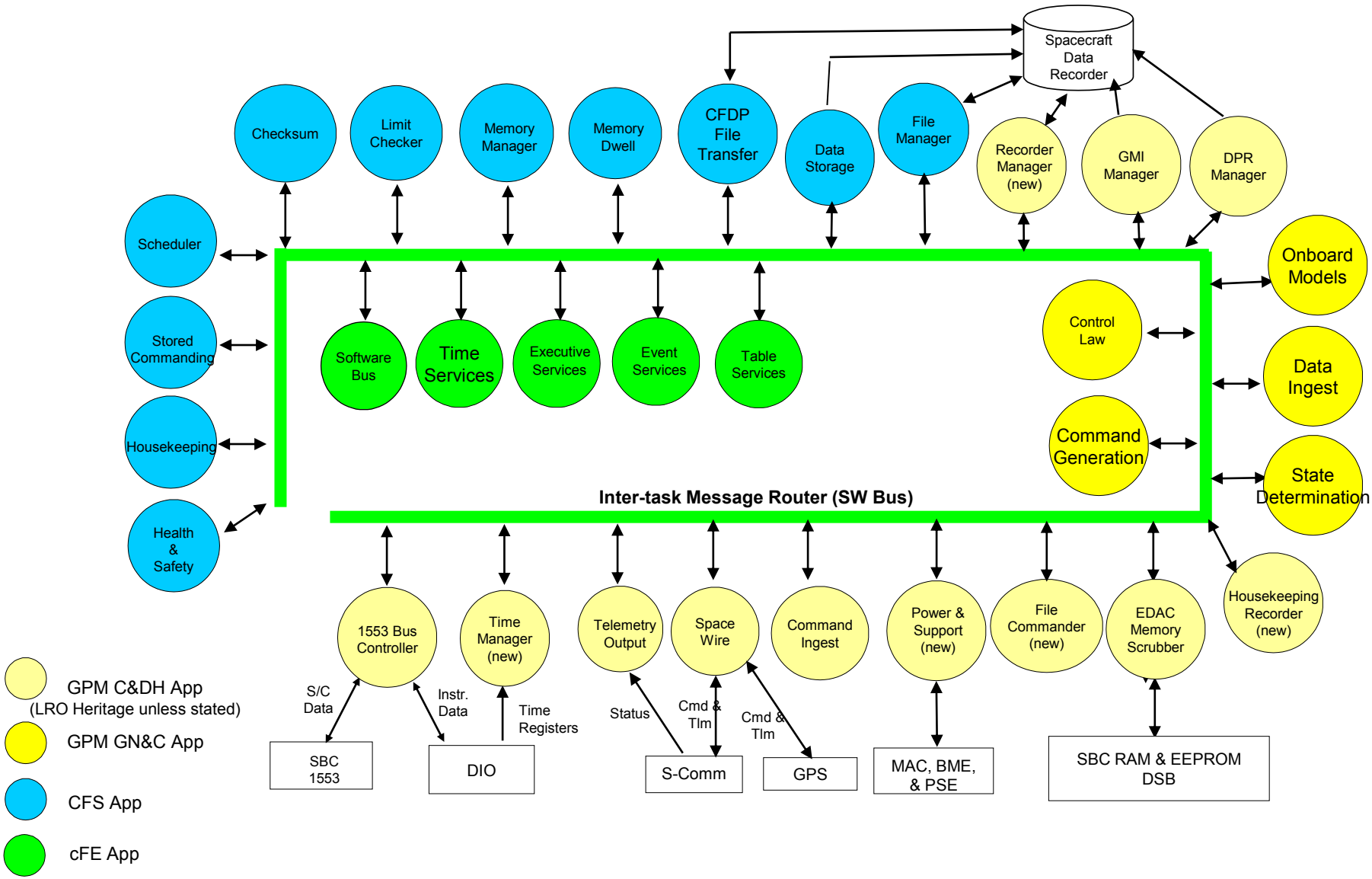
- Executes preloaded command sequences at predetermined absolute or relative time intervals
- Supports Absolute and Relative time-tagged sequences

CFS Component Metrics

Component	Version	Logical Lines of Code	Configuration Parameters
Core Flight Executive	6.3.2	12930	General: 17, Executive Service: 46 Event Service: 5, Software Bus: 29 Table Service: 10, Time Service: 32
CFDP	2.2.1	8559	33
Checksum	2.2.0	2873	15
Data Storage	2.3.0	2429	27
File Manager	2.3.1	1853	22
Health & safety	2.2.0	1531	45
Housekeeping	2.4.0	575	8
Limit Checker	2.2.1	2074	13
Memory Dwell	2.3.0	1035	8
Memory Manager	2.3.0	1958	25
Stored Commanding	2.3.0	2314	26
Scheduler	2.2.0	1164	19

- Two scopes of configuration parameters: mission or processor
- Configuration parameters span a large functional range from a simple default file name to a system behavioral definition like the time client/server configuration

Example Mission – Global Precipitation Measurement (GPM)



Example Mission – Global Precipitation Measurement (GPM)

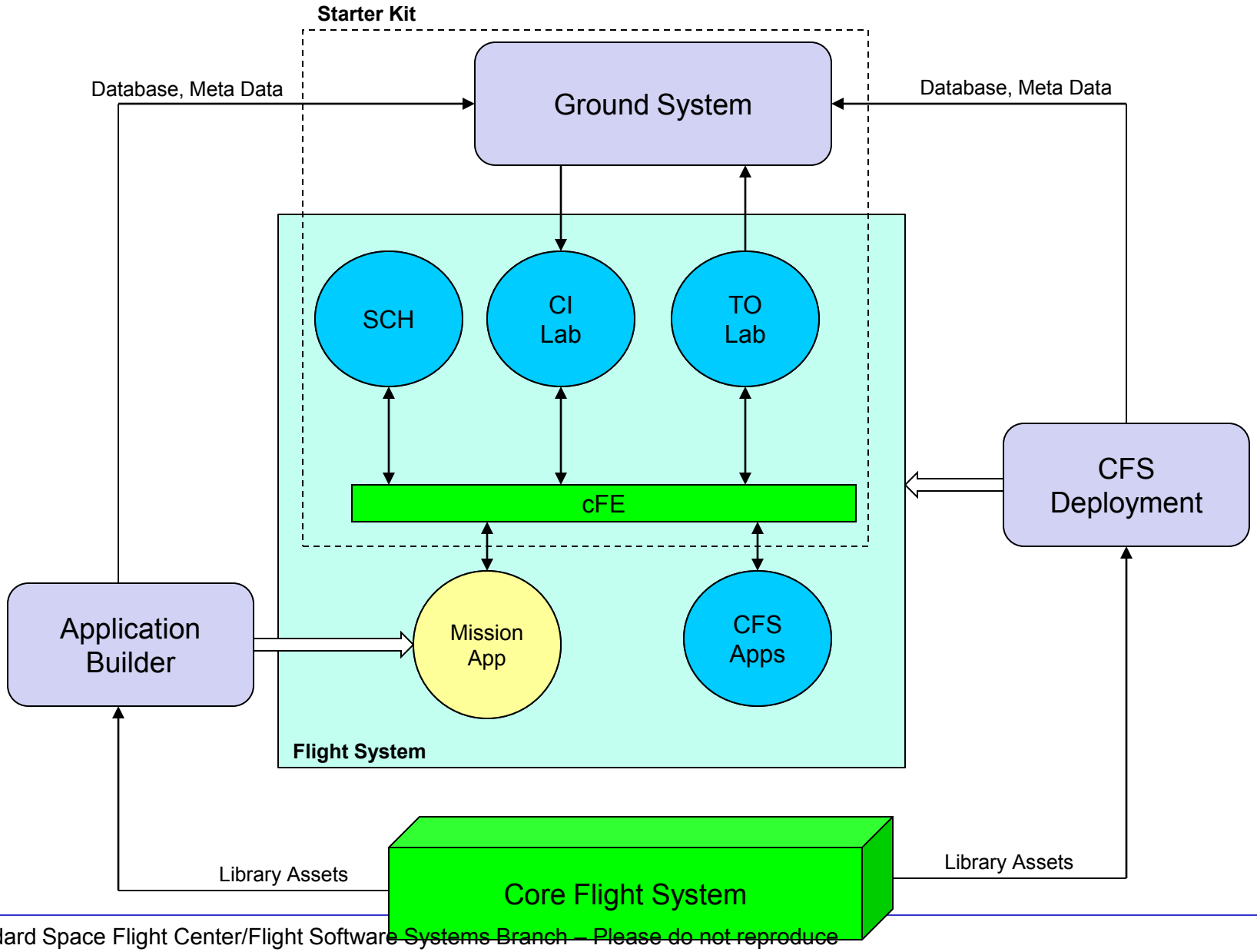
- **Noteworthy items**
  - \+ cFE was very reliable and stable
  - \+ Easy rapid prototyping with heritage code that was cFE compliant
  - \+ Layered architecture has allowed COTS lab to be maintained through all builds
  - – Addition of PSP changed build infrastructure midstream
- **Lines of Code Percentages:**

Source	Percentage
BAE	0.3
EEFS	1.7
OSAL	2.1
PSP	1.0
cFE	12.4
GNC Library	1.6
CFS Applications	23.5
Heritage Clone & Own	38.9
New Source	18.5

Next Steps

- **Establish a CFS community**
  - Define and implement a CFS governance model
  - Expand libraries at every layer
- **Continue IRADs**
  - Memory protection, multi-core, virtualization
- **Create a Starter Kit**
  - Make 3 applications available with the cFE: Scheduler, UDP-based CMD & TLM apps
  - Simple CMD & TLM ground system that is freely available and complies with a ground system independent database representation
- **Create CFS deployment Tools**
  - Tools to assist with CFS evaluation, configuration, and deployment
- **Create Application Builder Tools**
  - Translate models into applications (Ames did this for LADEE)
  - Aggregate reusable code modules into an application
- **Create Desktop Systems**
  - Closed loop systems that run on a desktop (JSC has done this for Morpheus)

Next Steps (cont)



Conclusion

- **Management Challenges**
  - Obtaining institutional funding for a collaborative community
  - Overcoming collaboration hurdles: ITAR, Software Release Processes, etc.
- **Technical Challenges**
  - Version control, bundling
  - Managing complexity: Product variants, configuration settings, etc.
  - New Technology Infusion
- **We've overcome challenges to get this far and we'll continue moving forward**
- **We welcome feedback!**
  - Susie Strege, susanne.l.strege@nasa.gov, 301-286-3829
  - David McComas, david.c.mccomas@nasa.gov ,301-286-9038

Backup Slides

References

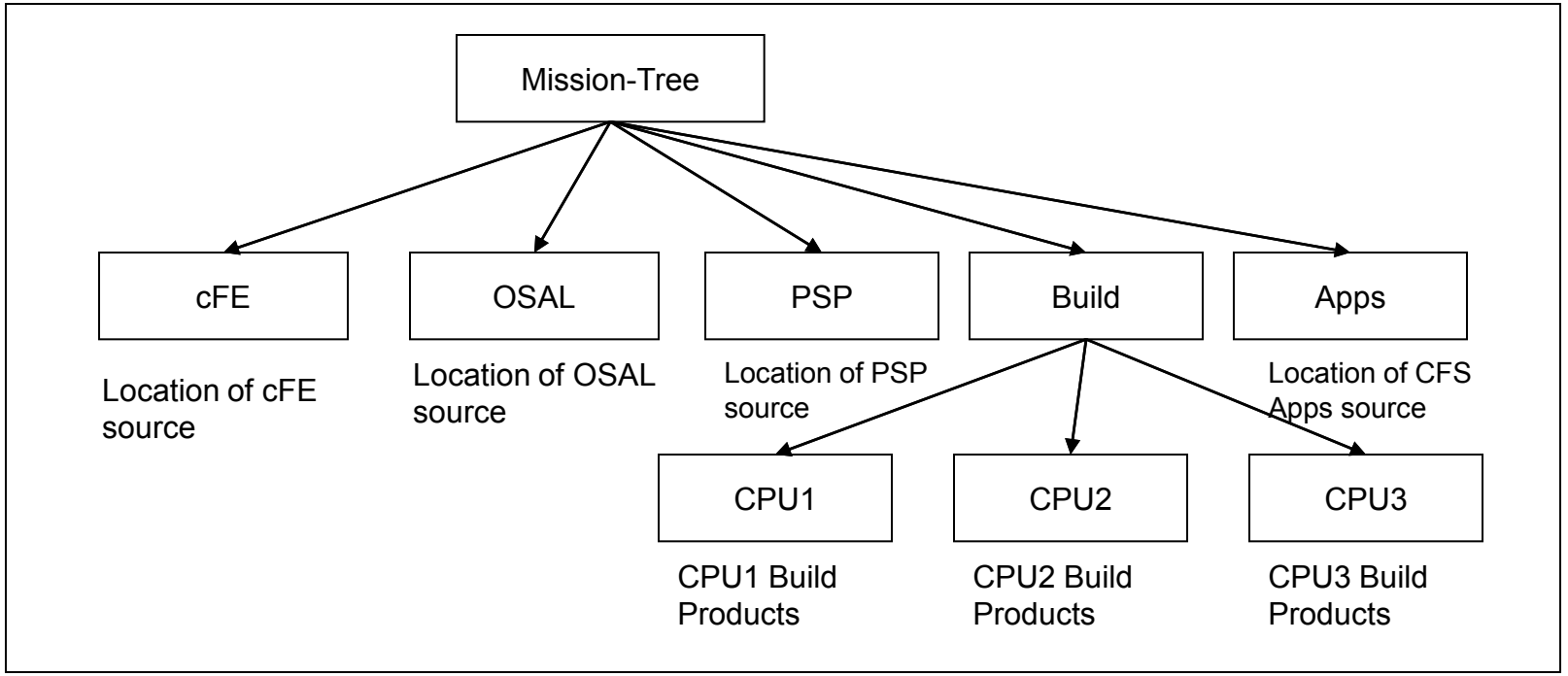
- **Open Source**
  - OSAL: http://osal.sf.net
  - cFE 6.2: http://sourceforge.net/projects/coreflightexec/files/cFE-6.1.1/
  - Goddard: http://opensource.gsfc.nasa.gov
  - NASA: http://code.nasa.gov/project/
  - OSAL and cFE migrating to Github in FY13

- **Goddard's Innovative Partnership Program Office**
  - http:// techtransfer.gsfc.nasa.gov
  - cfE POC: Enidia Santiago, enidia.santiago-arce-1@nasa.gov

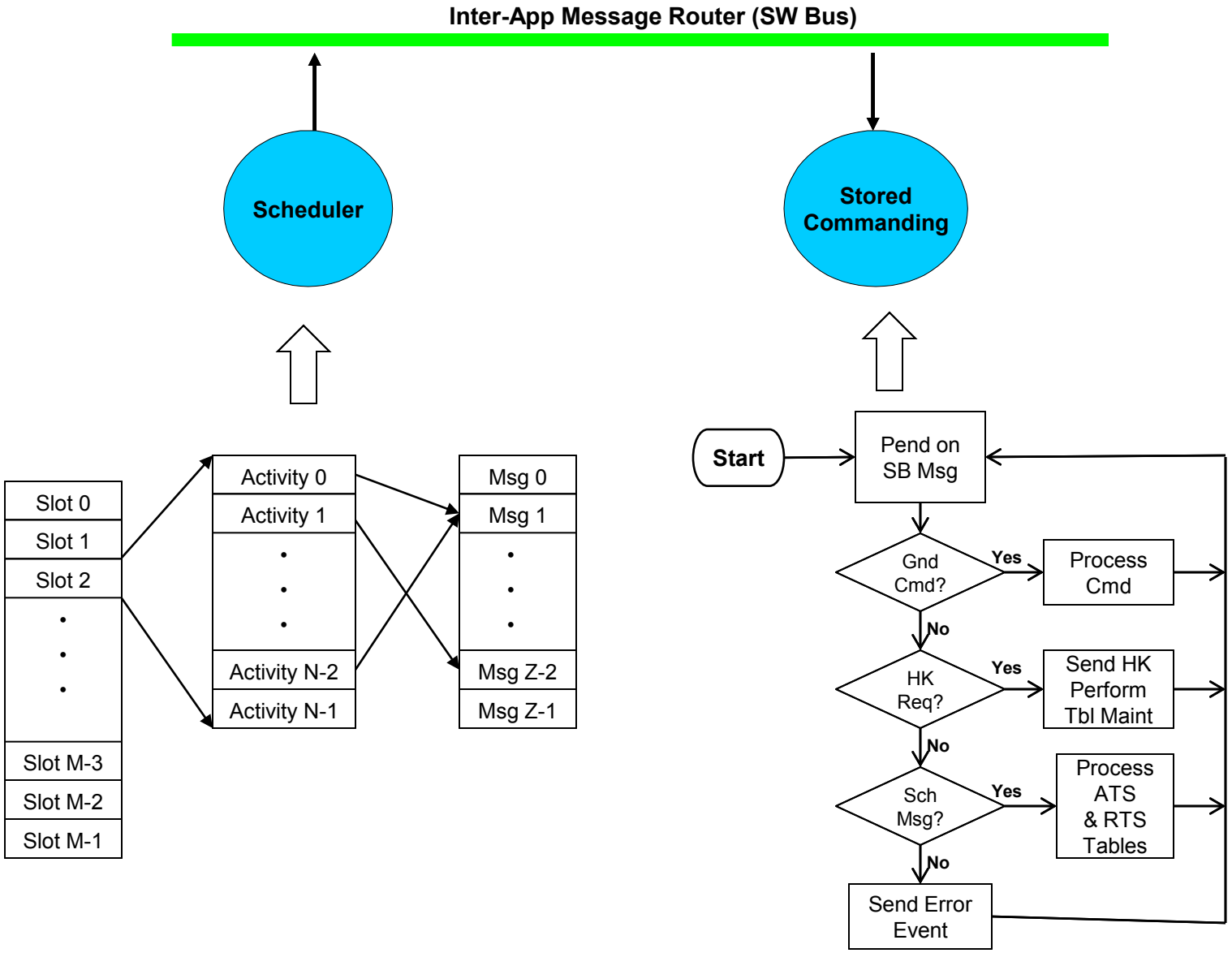
- **Publications**
  - Software Architecture Review Board (SARB) Review and Assessment of Goddard Space Flight Center's (GSFC's) core Flight Executive/Core Flight System (cFE/CFS), https://nen.nasa.gov/web/software/sarb
  - Verifying Architectural Design Rules of the Flight Software Product Line, Dharmalingam Ganesan, Mikael Lindvall, Chris Ackermann *Fraunhofer CESE,* http://www.fc-md.umd.edu/save

Development Environment

- The CFS has a complete development environment that is designed to manage:
  - Builds of images for multiple processors
  - Multiple processor architectures
  - Multiple operating systems
  - Different application loads on each processor
  - As little duplication of code as possible
  - Easy integration of mission patches due to Make search path



Scheduler


Inter-App Message Router (SW Bus)
Scheduler
Stored Commanding
Slot 0
Slot 1
Slot 2
Slot M-3
Slot M-2
Slot M-1
Activity 0
Activity 1
Activity N-2
Activity N-1
Msg 0
Msg 1
Msg Z-2
Msg Z-1
Start
Pend on SB Msg
Gnd Cmd?
Yes
Process Cmd
No
HK Req?
Yes
Send HK Perform Tbl Maint
No
Sch Msg?
Yes
Process ATS & RTS Tables
No
Send Error Event

Acronym List (1)

API	Application Programmer Interface
APL	Applied Physics Lab
ASIST	Advanced Spacecraft Integration and System Testing
ATS	Absolute Time Sequence
BC	Bus Controller
BT	Build Test
bps	bits-per seconds
Bps	Bytes-per seconds
BSP	Board Support Package
C&DH	Command and Data Handling
CCSDS	Consultative Committee for Space Data Systems
CDS	Critical Data Store
CESE	Center for Experimental Software Engineering
CFDP	CCSDS File Delivery Protocol
cFE	Core Flight Executive
CFS	Core Flight Software System
CM	Configuration Management
CMD	Command
COTS	Commercial Off The Shelf
CPST	TBD
cPCI	Compact PCI
CRC	Cyclic Redundancy Check
CS	Checksum
DMA	Direct Memory Access
DS	Data Storage
EEPROM	Electrically Erasable Programmable Read-Only Memory
EOF	End of File
ES	Executive Services
ESTO	TBD

Acronym List (3)

EVS	Event Services
FDC	Failure Detection and Correction
FM	File Management, Fault Management
FSW	Flight Software
GNC	Guidance Navigation and Control
GSFC	Goddard Space Flight Center
GOTS	Government Off The Shelf
GPM	Global Precipitation Measurement
GPS	Global Positioning System
Hi-Fi	High-Fidelity Simulation
HK	Housekeeping
HS	Health & Safety
HW	Hardware
Hz	Hertz
I&T	Integration and Test
ICD	Interface Control Document
IPP	Innovative Partnership Program Office
IRAD	Internal Research and Development
ITAR	International Traffic in Arms Regulations
ISR	Interrupt Service Routine
ITOS	Integration Test and Operations System
IV&V	Independent Verification and Validation
JHU	Johns Hopkins University
KORI	Korean Aerospace Research Institute
LADEE	Lunar Atmosphere and Dust Environment Explorer
LC	Limit Checker
LDS	Local Data Storage
LRO	Lunar Reconnaissance Orbiter
Mbps	Megabits-per seconds

Acronym List (5)

MD	Memory Dwell
MET	Mission Elapsed Timer
MM	Memory Manager
MS	Memory Scrub
NACK	Negative-acknowledgement
NASA	National Aeronautics Space Agency
NESC	NASA Engineering and Safety Center
NOOP	No Operation
OS	Operating System
OSAL	Operating System Abstraction Layer
PCI	Peripheral Component Interconnect
PSP	Platform Support Package
RAM	Random-Access Memory
RM	Recorder Manager
ROM	Read-Only Memory
RT	Remote Terminal
R/T	Real-time
RTOS	Real-Time Operating System
RTS	Relative Time Sequence
SARB	Software Architecture Review Board
S/C	Spacecraft
SB	Software Bus
SBC	Single-Board Computer
SC	Stored Command
SCH	Scheduler
S-COMM	S-Band Communication Card
SDO	Solar Dynamic Observatory
SDR	Spacecraft Data Recorder

Acronym List (7)

SIL	Simulink Interface Layer
SpW	Spacewire
SRAM	Static RAM
SSR	Solid State Recorder
STCF	Spacecraft Time Correlation Factor
SUROM	Start-Up Read-Only Memory
SW	Software, Spacewire
TAI	International Atomic Time
TBD	To be determined
TBL	Table Services
TLM	Telemetry
TDRS	Tracking Data Relay Satellite
TM	Time Manager
TO	Telemetry Output
TRMM	Tropical Rainfall Measuring System
UART	Universal Asynchronous Receiver/Transmitter
UDP	User Datagram Protocol
UMD	University of Maryland
UT	Unit Test
UTC	Coordinated Universal Time
VCDU	Virtual Channel Data Unit
XB	External Bus
XBI	Instrument 1553 External Bus
XBS	Spacecraft 1553 External Bus